We claim:

 A method for protecting entry addresses, the method which comprises:

identifying a permissible entry address by using a correlation of data, wherein the data are not provided within a same individual instruction; and

storing, in a memory cell, an address of a correlated data item one of directly before and directly after the permissible entry address.

- 2. The method according to claim 1, which comprises storing, in the memory cell, a reference to a data entry in a protected list of legal entry addresses one of directly before and directly after the permissible entry address.
- The method according to claim 1, which comprises directly jumping to the permissible entry address.
- 4. The method according to claim 1, which comprises automatically checking whether the correlation of data is satisfied for a respective entry address, when a function call is carried out.

5. A method for protecting entry addresses, the method which comprises:

identifying a permissible entry address by using a correlation of data, wherein the data are not provided within a same individual instruction; and

providing the correlation of data as a correlation with program data in non-reserved memory areas.

6. The method according to claim 5, which comprises:

providing program instructions not exceeding a given maximum number n of bytes, n being an integer number; and

providing a specific no-operation code for avoiding random correlations.

- 7. The method according to claim 5, which comprises providing the correlation of data as a correlation between code data items, the code data items being at least n bytes away from one another, n being an integer number.
- The method according to claim 6, which comprises providing the correlation of data as a correlation between code data

items, the code data items being at least n bytes away from one another.

9. The method according to claim 5, which comprises:

providing a specific byte sequence which cannot occur within a regular code; and

protecting the permissible entry address by inserting the specific byte sequence.

- 10. The method according to claim 9, which comprises using a specific no-operation code as the specific byte sequence.
- 11. The method according to claim 5, which comprises jumping directly to the permissible entry address.